Local Codes with Addition Based Repair

Han Mao Kiah[†], Son Hoang Dau[*], Wentu Song[‡], Chau Yuen[§]
[†]School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore
[*‡§]Singapore University of Technology and Design, Singapore
Emails: [†]*hmkiah*@ntu.edu.sg, {[*]*sonhoang_dau*, [‡]*wentu_song*, [§]*yuenchau*}@sutd.edu.sg

Abstract

We consider the complexities of repair algorithms for locally repairable codes and propose a class of codes that repair single node failures using addition operations only, or *codes with addition based repair*. We construct two families of codes with addition based repair. The first family attains distance one less than the Singleton-like upper bound, while the second family attains the Singleton-like upper bound.

1. Introduction

Motivated by practical implementations of erasure coding in large-scale distributed storage systems, *locality* was identified as an important metric of study independently by Gopalan *et al.* [1], Oggier and Datta [2], and Papailiopoulos and Dimakis [3]. Generally, the locality of a node i refers to the number of other nodes that needs to be accessed in order to recover node i. We refer to the latter set of nodes as the *repair set* for i and observe that having small values of locality allows for the fast repair of single node failures.

This observation has impelled the study and construction of erasure codes with low locality and high minimum distance. Extending the argument from the classical Singleton bound, Gopalan *et al.* [1] established a relationship between the parameters: code length n, code dimension k, locality r and minimum distance d (see Theorem 2.1). Since then, families of optimal codes attaining this bound were constructed (see [4]–[8]).

However, the complexity of the repair algorithms for these code families is often not explicitly discussed. Nevertheless, we observe that the repair algorithms usually involve multiplication and division in some finite field – operations that are costly to implement in software and hardware. This cost is exacerbated when the underlying finite field is of a large order. In the literature of coding for disk arrays, this computational complexity is reduced by replacing finite field arithmetic by simple bit-wise operations. For example, Blaum and Roth [9] proposed a construction of array codes based on the ring of polynomials with binary coefficients. More recently, Hou *et al.* [10] and Shum *et al.* [11] proposed classes of regenerating codes that utilize only XOR operations and bit-wise shifts.

Following this line of study, we propose a simple and efficient method of repair: when a node fails, we compute the sum of the nodes in its repair set to obtain S, and replace the failed node by $-S$. Hence, our repair algorithm uses *only addition* operations in a finite field and when the field is a binary extension field (as with most storage applications), it utilizes *only XOR* operations. In this paper, we construct two families of linear codes with good locality properties that utilizes this algorithm to repair *all* its nodes. The first family achieves distance one less than the value in (1), while the second family achieves exactly the distance, albeit under certain conditions. These results hence indicate that the extra requirement of addition based repair may be achieved *without significant loss in erasure correcting capability*.

Example 1.1. Consider a linear code $\mathcal{C}$ of length twelve and dimension six defined over $\mathbb{F}_{13}$. Let the information symbols be $x_1, x_2, \ldots, x_6$ and the generator matrix of $\mathcal{C}$ be

$$\mathbf{G} = \left(\begin{array}{cccc|cccc|cccc} 1 & 0 & 0 & 12 & 0 & 0 & 0 & 0 & 7 & 8 & 10 & 1 \\ 0 & 1 & 0 & 12 & 0 & 0 & 0 & 0 & 8 & 2 & 5 & 11 \\ 0 & 0 & 1 & 12 & 0 & 0 & 0 & 0 & 5 & 3 & 12 & 6 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 12 & 1 & 6 & 2 & 4 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 12 & 5 & 7 & 8 & 6 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 12 & 7 & 11 & 9 & 12 \end{array}\right).$$

Part of this work was completed when Han Mao Kiah visited Singapore University of Technology and Design.

We label the nodes with $1, 2, \ldots, 12$ and consider the information stored in the last three nodes 9, 10, 11, 12:

$$\begin{aligned} c_9 &= 7x_1 + 8x_2 + 5x_3 + 1x_4 + 5x_5 + 7x_6, \\ c_{10} &= 8x_1 + 2x_2 + 3x_3 + 6x_4 + 7x_5 + 11x_6, \\ c_{11} &= 10x_1 + 5x_2 + 12x_3 + 2x_4 + 8x_5 + 9x_6, \\ c_{12} &= 1x_1 + 11x_2 + 6x_3 + 4x_4 + 6x_5 + 12x_6. \end{aligned}$$

Observe that $c_9 = -(c_{10} + c_{11} + c_{12})$. Hence, when node 9 fails, we repair the node by contacting nodes 10, 11 and 12, and compute c_9 via addition operations over $\mathbb{F}_{13}$. We check that this code has distance six, thus attaining the upper bound (1).

The paper is organized as follows. We first describe our main results in Section 2 and provide the explicit construction of these code families in Section 3. In Section 4, we compare our codes with known families of codes with good locality properties., namely, Pyramid codes [12] and Tamo-Barg codes [8]. In particular, we provide an analysis on the repair complexities for these codes.

2. Main Results

Let $[n] \triangleq \{1, 2, \ldots, n\}$. Consider a linear $[n, k, d]$ code $\mathcal{C}$ over $\mathbb{F}_q$ of length n with k *information* nodes, $n-k$ *parity check* nodes and distance d. A node $i \in [n]$ has *locality* r_i if there exists a *repair set* $J_i = \{j_1, j_2, \ldots, j_{r_i}\} \subseteq [n] \setminus \{i\}$ of size r_i and a *repair function* $\phi_i : \mathbb{F}_q^{r_i} \to \mathbb{F}_q$ such that $\phi_i(c_{j_1}, c_{j_2}, \ldots, c_{j_{r_i}}) = c_i$ for all $\mathbf{c} \in \mathcal{C}$. The code $\mathcal{C}$ has *information locality* r if all information nodes have locality at most r, and has *all-symbol locality* r if all nodes have locality at most r.

We are interested in repair functions that utilizes only addition operations. Formally, a code $\mathcal{C}$ has *addition based repair* if for each node $i \in [n]$, the repair function defined on the repair set $J_i = \{j_1, j_2, \ldots, j_{r_i}\}$ is given by

$$\phi_i(c_{j_1}, c_{j_2}, \ldots, c_{j_{r_i}}) = -(c_{j_1} + c_{j_2} + \cdots + c_{j_{r_i}}).$$

Hence, our objective is to construct a code with addition based repair and good locality properties, while maximizing its minimum distance. To benchmark the performance of our codes, we use the following Singleton-like upper bound on the distance of codes with information locality r.

Theorem 2.1 ([1, Th. 5]). Consider an $[n, k, d]$ linear code with information locality r. Then

$$d \le n - \left\lceil \frac{k}{r} \right\rceil - k + 2. \tag{1}$$

Observe that the above bound applies to linear codes with all-symbol locality r and in the same paper, Gopalan *et al.* provided a structure theorem for these codes [1]. In particular, they demonstrated that under certain conditions, codes with all-symbol locality r attaining the upper bound (1) were not possible. Song *et al.* then further extended this result [6].

Theorem 2.2 ([1, Cor. 10], [6, Th. 10]). Suppose $0 < r < k \le n$, $r|k$ and $r+1 \nmid n$. Then there exists no $[n, k, d]$ linear code with all-symbol locality r with $d = n - k - k/r + 2$.

We present our key results in Theorem 2.3 and 2.5, and defer their proofs to Section 3. Our first construction provides a family of linear codes with addition based repair, information locality r and attains distance one less than the value in (1).

Theorem 2.3 (Construction I). Fix $n < q$, $0 < r < k \le n$ and $r|k$. Let $t = n - k - k/r$. Then there exists an $[n, k, t+1]$ linear code over $\mathbb{F}_q$ with addition based repair and information locality r, where k/r parity symbols have locality r, and t remaining parity symbols have locality $\min\{t-1, k\}$.

Observe that in the case where $t \le r+1$, the code given by Theorem 2.3 has all-symbol locality r. Furthermore, when $t < r+1$, we have that $r+1 \nmid n$ and so, Theorem 2.2 applies and yields the following corollary.

Corollary 2.4. Fix $n < q$, $0 < r < k \le n$ and $r|k$. Suppose that $t = n - k - k/r \le r+1$. There exists an $[n, k, t+1]$ linear code over $\mathbb{F}_q$ with addition based repair and all-symbol locality r. Furthermore, if $t < r+1$, then the code is optimal in terms of distance.

Our second construction provides a family of linear codes with addition based repair, all-symbol locality r and attains the distance in (1).

Theorem 2.5 (Construction II)**.** Fix $n < q$, $0 < r < k \leq n$ and $r|k$. Suppose that $n|r+1$, $r+1|q-1$, and $t = n - k - k/r$. Then there exists an $[n, k, t+2]$ linear code over $\mathbb{F}_q$ with addition based repair and all-symbol locality r. By (1), the code is optimal in terms of distance.

3. Code Constructions

For the rest of the paper, we fix $n < q$, $0 < r < k \leq n$, $r|k$ and let $t = n - k - k/r$ and $m = k/r$. In this section, we provide two constructions of codes with addition based repair.

(I) The first code construction simply requires $n < q$. Codes from this family have information locality r and has distance $t+1$.

(II) Codes in the second family have $r+1|n$ and we require $r+1|q-1$. These codes have all-symbol locality r and distance $t+2$.

A. *Construction I*

To construct codes for Theorem 2.3, we produce a $k \times n$ generator matrix $\mathbf{G}$ of the form

$$\left(\begin{array}{c|c|c|c|cccc}
 & & & & g_0^{(1)} & g_1^{(1)} & \cdots & g_{t-1}^{(1)} \\
\mathbf{M} & \mathbf{0} & \cdots & \mathbf{0} & g_0^{(2)} & g_1^{(2)} & \cdots & g_{t-1}^{(2)} \\
 & & & & \vdots & \vdots & \ddots & \vdots \\
 & & & & g_0^{(r)} & g_1^{(r)} & \cdots & g_{t-1}^{(r)} \\ \hline
 & & & & g_0^{(r+1)} & g_1^{(r+1)} & \cdots & g_{t-1}^{(r+1)} \\
\mathbf{0} & \mathbf{M} & \cdots & \mathbf{0} & g_0^{(r+2)} & g_1^{(r+2)} & \cdots & g_{t-1}^{(r+2)} \\
 & & & & \vdots & \vdots & \ddots & \vdots \\
 & & & & g_0^{(2r)} & g_1^{(2r)} & \cdots & g_{t-1}^{(2r)} \\ \hline
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \hline
 & & & & g_0^{(k-r+1)} & g_1^{(k-r+1)} & \cdots & g_{t-1}^{(k-r+1)} \\
\mathbf{0} & \mathbf{0} & \cdots & \mathbf{M} & g_0^{(k-r+2)} & g_1^{(k-r+2)} & \cdots & g_{t-1}^{(k-r+2)} \\
 & & & & \vdots & \vdots & \ddots & \vdots \\
 & & & & g_0^{(k)} & g_1^{(k)} & \cdots & g_{t-1}^{(k)}
\end{array}\right),$$

where $\mathbf{M}$ is the $r \times (r+1)$ matrix $(\mathbf{I}_r | -\mathbf{j}_r)$. Here, $\mathbf{I}_\ell$ and $\mathbf{j}_\ell$ denote the identity matrix of dimension ℓ and the all-ones vector of length ℓ, respectively.

Clearly, $\mathbf{G}$ has full rank and hence, it remains to find the elements $g_j^{(i)}$ for $i \in [k]$ and $0 \leq j \leq t-1$ such that $\mathbf{G}$ generates a linear code $\mathcal{C}$ of distance $t+1$ with addition based repair. To this end, we demonstrate that $\mathcal{C}$ is a punctured subcode of a cyclic code of length $q-1$ with the appropriate distance.

Specifically, we identify a vector $(c_0, c_1, \ldots, c_{q-2})$ of length $q-1$ with the polynomial $c(x) = \sum_{j=0}^{q-2} c_j x^j$ in the ring $\mathbb{F}_q^{q-1}/\langle x^{q-1} - 1\rangle$. Under this mapping of vectors to polynomials, let the rows of $\mathbf{G}$ be mapped to polynomials $c^{(1)}(x)$, $c^{(2)}(x)$,..., $c^{(k)}(x)$, and so, the code $\mathcal{C}$ is identified with the vector space generated by $\{c^i(x) : i \in [k]\}$. Note that each codeword of $\mathcal{C}$ has length $n \leq q-1$. However, we can regard them as vectors of length $q-1$ by appending $q-1-n$ zeros to the right of each codeword and this operation does not affect the polynomial representation of the codewords. The next lemma provides sufficient conditions for $c^{(i)}(x)$ to generate a code with addition based repair and distance $t+1$.

Lemma 3.1. Let ω be the primitive element in $\mathbb{F}_q$. Suppose that $\mathcal{C}$ is generated by $\{c^{(i)}(x) : i \in [k]\}$ as defined above. If $c^{(i)}(x)$ has roots at $1, \omega, \ldots, \omega^{t-1}$ for all $i \in [k]$, then $\mathcal{C}$ has addition based repair and distance $t+1$.

To prove this lemma, we employ the BCH bound on minimum distance of a cyclic code. We first recall some notation from [13, Ch. 7]. Let $\mathcal{C}$ be a cyclic code of length $q-1$ over $\mathbb{F}_q$. An element $\alpha \in \mathbb{F}_q$ is called a *zero* of $\mathcal{C}$

if $c(\alpha) = 0$ for every codeword $c(x) \in \mathcal{C}$. Let Z be the set of all zeros of $\mathcal{C}$. The polynomial $g(x) \triangleq \prod_{\alpha \in Z}(x-\alpha)$ is called the *generator polynomial* of $\mathcal{C}$. Then $c(x) \in \mathcal{C}$ if and only if $g(x)|c(x)$.

Theorem 3.2 (BCH bound)**.** Let ω be the primitive element of $\mathbb{F}_q$ and t be an integer. The cyclic code with the generator polynomial $(x-1)(x-\omega)\cdots(x-\omega^{t-1})$ has minimum distance at least $t+1$.

Proof of Lemma 3.1: For the distance properties, we show that $\mathcal{C}$ is a subcode of a cyclic code with generator polynomial $g(x) = (x-1)(x-\omega)\cdots(x-\omega^{t-1})$. Then the distance of $\mathcal{C}$ is immediate from the BCH bound.

Indeed, for $i \in [k]$, since $c^{(i)}(x)$ has roots at $1, \omega, \ldots, \omega^{t-1}$, $g(x)$ divides $c^{(i)}(x)$. Hence, $g(x)$ divides all polynomials in the span of $\{c^{(i)}(x) : i \in [k]\}$, or, $\mathcal{C}$, and, the claim follows.

To demonstrate that $\mathcal{C}$ has addition based repair, we partition the first $n-t$ coordinates into m groups of $r+1$ nodes and place the last t coordinates into one group. Therefore, it suffices to show that the sum of the symbols in each group is zero for a typical codeword in $\mathcal{C}$. Equivalently, we verify this fact for all rows in the generator matrix $\mathbf{G}$.

Now, the claim is clear for the first m groups from the description of $\mathbf{G}$. For the last group of t symbols, consider $i \in [k]$. Then $c^{(i)}(x) = x^{t_1} - x^{t_2} + x^{n-t}\sum_{j=0}^{t-1} g_j^{(i)} x^j$ for some $0 \le t_1 < t_2 \le n-t-1$. Since $c^{(i)}(1) = 0$, we have $\sum_{j=0}^{t-1} g_j^{(i)} = 0$, as desired. ■

Therefore, to complete the proof of Theorem 2.3, we show that $g_j^{(i)}$ exist for $i \in [k]$ and $0 \le j \le t-1$ and they can be found by solving k linear systems, each involving t equations in t variables. Specifically, fix $i \in [k]$ and let $c^{(i)}(x) = x^{t_1} - x^{t_2} + x^{n-t}\sum_{j=0}^{t-1} g_j^{(i)} x^j$ for some $0 \le t_1 < t_2 \le n-t-1$. Then $c^{(i)}(x)$ having roots at $1, \omega, \ldots, \omega^{t-1}$ is equivalent to the following linear system:

$$\begin{pmatrix} 1 & 1 & \cdots & 1 \\ \omega^{n-t} & \omega^{n-t+1} & \cdots & \omega^{n-1} \\ \vdots & \vdots & \ddots & \vdots \\ \omega^{(t-1)(n-t)} & \omega^{(t-1)(n-t+1)} & \cdots & \omega^{(t-1)(n-1)} \end{pmatrix} \mathbf{g} = \mathbf{w},$$

where $\mathbf{g} = (g^{(i)})_{j=0}^{t-1}$ and $\mathbf{w} = (\omega^{t_1 j} - \omega^{t_2 j})_{j=0}^{t-1}$.

Observe that the matrix representing the system of equations is Vandermonde. Hence, we have a unique solution for $g_j^{(i)}$ and this completes the proof of Theorem 2.3.

Example 3.1. Let $n = 11$, $q = 13$, $r = 3$ and $k = 6$. To compute the matrix $\mathbf{G}$, we solve six linear systems, each involving three equations in three unknowns. Hence,

$$\mathbf{G} = \left(\begin{array}{cccc|cccc|ccc} 1 & 0 & 0 & 12 & 0 & 0 & 0 & 0 & 9 & 10 & 7 \\ 0 & 1 & 0 & 12 & 0 & 0 & 0 & 0 & 12 & 4 & 10 \\ 0 & 0 & 1 & 12 & 0 & 0 & 0 & 0 & 10 & 4 & 12 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 12 & 11 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 12 & 9 & 7 & 10 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 12 & 11 & 8 & 7 \end{array}\right),$$

and $\mathbf{G}$ generates an $[11, 6, 4]$ linear code $\mathcal{C}$ with addition based repair. Nodes 1 to 8 have locality three, while nodes 9, 10, 11 have locality two. Hence, $\mathcal{C}$ has all-symbol locality three. Since no $[11, 6, 5]$ linear code exists with all-symbol locality three by Theorem 2.2, we conclude that $\mathcal{C}$ is optimal. □

Remark 1. We consider alternative approaches to the problem and point out certain pitfalls. Suppose there exists an $[k+t-1, k, t]$ MDS code. One common approach in the literature is to partition the $k+t-1$ nodes into $m+1$ groups (m groups of size r and one group of size $t-1$), and re-encode each group with either a $[r+1, r, 2]$ or $[t, t-1, 2]$ MDS code. Equivalently, we add another $m+1$ nodes to obtain a code of length n with the desired locality properties. The question is then whether the resulting code has addition based repair and the desired distance.

(i) One obvious way to guarantee addition based repair is to re-encode each group with MDS codes that have $(\mathbf{I}_r | -\mathbf{j}_r)$ and $(\mathbf{I}_{t-1} | -\mathbf{j}_{t-1})$ as their generator matrices.

Unfortunately, this method does not ensure that the resulting code is of distance $t+1$. Consider the following generator matrix of a $[7,4,4]$ MDS code over $\mathbb{F}_7$,

$$\left(\begin{array}{cc|cc|ccc} 1 & 0 & 0 & 0 & 1 & 1 & 4 \\ 0 & 1 & 0 & 0 & 1 & 2 & 3 \\ 0 & 0 & 1 & 0 & 2 & 1 & 3 \\ 0 & 0 & 0 & 1 & 2 & 6 & 5 \end{array}\right).$$

Adding three nodes as above yields the generator matrix,

$$\left(\begin{array}{ccc|ccc|cccc} 1 & 0 & 6 & 0 & 0 & 0 & 1 & 1 & 4 & 1 \\ 0 & 1 & 6 & 0 & 0 & 0 & 1 & 2 & 3 & 1 \\ 0 & 0 & 0 & 1 & 0 & 6 & 2 & 1 & 3 & 1 \\ 0 & 0 & 0 & 0 & 1 & 6 & 2 & 6 & 5 & 1 \end{array}\right).$$

However, the distance of the new code remains as four (distance between the first two rows is four).

(ii) When $t = r + 1$, a more elaborate re-encoding was proposed by Tamo *et al.* [5]. The initial MDS code in their method is chosen to be a Reed Solomon code defined over $\mathbb{F}_q$, while the groups are re-encoded using carefully chosen MDS codes defined over an extension field $\mathbb{F}_{q^{k+1}}$. Tamo *et al.* then demonstrated that the distance of the new code is increased to $t + 2$, thus achieving the bound (1).

However, the repair algorithm for the code involves multiplications over a finite field. This complexity is further increased by the fact that the underlying field is of the order q^{k+1}. In general, multiplication over a binary extension field of order 2^s involves $O(s^{\log_2 3})$ bit operations (see for example, [14]). Thus, in the scheme of Tamo *et al.*, when the code is defined over $\mathbb{F}_{q^{k+1}}$, the number of bit operations involved in the repair is increased by a factor of $(k+1)^{\log_2 3}$.

Remark 2. Recall below a code construction from Gopalan *et al.* [1] with similar parameters as the codes in Corollary 2.4.

Theorem 3.3 ([1, Th. 15])**.** Let $0 < r < k \le n$ and $t = n - k - k/r < r + 1$. Suppose that $q > kn^k$. There exists a systematic $[n, k, t+2]_q$ code $\mathcal{C}$ of information locality r, where k/r parity symbols have locality r, and t remaining parity symbols have locality $k - (k/r - 1)(t - 1)$.

We compare our codes constructed in Corollary 2.4 and the codes in Theorem 3.3 in Table I. Observe that our codes besides having addition based repair, also have smaller locality values and are defined over a significantly smaller field. However, the codes by Gopalan *et al.* have distance one more than our codes.

	Our code	Gopalan *et al.* code
Addition based repair	Yes	No
Locality of $k + k/r$ nodes	r	r
Locality of t nodes	$t - 1$	$t - 1 + (r + 1 - t)k/r$
Field size	$\mathcal{O}(n)$	$\mathcal{O}(kn^k)$
Minimum distance	$t + 1$	$t + 2$

TABLE I
COMPARISON OF CODE FROM COROLLARY 2.4 AND CODE FROM THEOREM 3.3 (BY GOPALAN *et al.*).

B. Construction II

We construct codes that satisfy the conditions in Theorem 2.5. Here, we assume further $r + 1 | n$ and let $n = (k/r + \ell)(r + 1) = (m + \ell)(r + 1)$, in other words, $t = \ell r + \ell$. Also suppose that $r + 1 | q - 1$ and let ω be a primitive element of $\mathbb{F}_q$. Define $\alpha \triangleq \omega^{(q-1)/(r+1)}$, and so, α has order $r + 1$.

In lieu of a generator matrix, we construct a *parity-check matrix* for our code. Observe that $n - k = m + \ell + \ell r$. Hence, we consider the $(m + \ell + \ell r) \times n$ matrix $\mathbf{H}$ of the form

$$
\mathbf{H} \triangleq \left(\begin{array}{cccc|cccc|c|cccc}
1 & 1 & \cdots & 1 & 0 & 0 & \cdots & 0 & \cdots & 0 & 0 & \cdots & 0 \\
0 & 0 & \cdots & 0 & 1 & 1 & \cdots & 1 & \cdots & 0 & 0 & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & \cdots & 1 & 1 & \cdots & 1 \\
\hline
\multicolumn{4}{c|}{\mathbf{v}_0^{(1)}} & \multicolumn{4}{c|}{\mathbf{v}_1^{(1)}} & \cdots & \multicolumn{4}{c}{\mathbf{v}_{m+\ell-1}^{(1)}} \\
\multicolumn{4}{c|}{\mathbf{v}_0^{(2)}} & \multicolumn{4}{c|}{\mathbf{v}_1^{(2)}} & \cdots & \multicolumn{4}{c}{\mathbf{v}_{m+\ell-1}^{(2)}} \\
\multicolumn{4}{c|}{\vdots} & \multicolumn{4}{c|}{\vdots} & \cdots & \multicolumn{4}{c}{\vdots} \\
\multicolumn{4}{c|}{\mathbf{v}_0^{(r)}} & \multicolumn{4}{c|}{\mathbf{v}_1^{(r)}} & \cdots & \multicolumn{4}{c}{\mathbf{v}_{m+\ell-1}^{(r)}} \\
\hline
\multicolumn{4}{c|}{\mathbf{v}_0^{(r+2)}} & \multicolumn{4}{c|}{\mathbf{v}_1^{(r+2)}} & \cdots & \multicolumn{4}{c}{\mathbf{v}_{m+\ell-1}^{(r+2)}} \\
\multicolumn{4}{c|}{\mathbf{v}_0^{(r+3)}} & \multicolumn{4}{c|}{\mathbf{v}_1^{(r+3)}} & \cdots & \multicolumn{4}{c}{\mathbf{v}_{m+\ell-1}^{(r+3)}} \\
\multicolumn{4}{c|}{\vdots} & \multicolumn{4}{c|}{\vdots} & \cdots & \multicolumn{4}{c}{\vdots} \\
\multicolumn{4}{c|}{\mathbf{v}_0^{(2r+1)}} & \multicolumn{4}{c|}{\mathbf{v}_1^{(2r+1)}} & \cdots & \multicolumn{4}{c}{\mathbf{v}_{m+\ell-1}^{(2r+1)}} \\
\hline
\multicolumn{4}{c|}{\vdots} & \multicolumn{4}{c|}{\vdots} & \ddots & \multicolumn{4}{c}{\vdots} \\
\hline
\multicolumn{4}{c|}{\mathbf{v}_0^{(\ell r+\ell-r)}} & \multicolumn{4}{c|}{\mathbf{v}_1^{(\ell r+\ell-r)}} & \cdots & \multicolumn{4}{c}{\mathbf{v}_{m+\ell-1}^{(\ell r+\ell-r)}} \\
\multicolumn{4}{c|}{\mathbf{v}_0^{(\ell r+\ell-r+1)}} & \multicolumn{4}{c|}{\mathbf{v}_1^{(\ell r+\ell-r+1)}} & \cdots & \multicolumn{4}{c}{\mathbf{v}_{m+\ell-1}^{(\ell r+\ell-r+1)}} \\
\multicolumn{4}{c|}{\vdots} & \multicolumn{4}{c|}{\vdots} & \cdots & \multicolumn{4}{c}{\vdots} \\
\multicolumn{4}{c|}{\mathbf{v}_0^{(\ell r+\ell-1)}} & \multicolumn{4}{c|}{\mathbf{v}_1^{(\ell r+\ell-1)}} & \cdots & \multicolumn{4}{c}{\mathbf{v}_{m+\ell-1}^{(\ell r+\ell-1)}}
\end{array}\right),
$$

where

$$\mathbf{v}_i^{(j)} = ((\omega^i)^j, (\omega^i\alpha)^j, \ldots, (\omega^i\alpha^r)^j) \text{ for } 0 \le i \le m+\ell-1, 0 \le j \le n-1.$$

Observe that $0 \le j \le n-1$, the vector $\left(\mathbf{v}_i^{(j)}\right)_{i=0}^{m+\ell-1}$ is the evaluation of the polynomial x^j with evaluation points $\{\omega^i\alpha^{i_1} : 0 \le i \le m+\ell-1, 0 \le i_1 \le r\}$. Therefore, the set of n vectors $\mathbf{V} \triangleq \left\{\left(\mathbf{v}_i^{(j)}\right)_{i=0}^{m+\ell-1} : 0 \le j \le n-1\right\}$ is linearly independent.

We first demonstrate the following lemma.

Lemma 3.4. For $0 \le i \le m+\ell-1, 0 \le j \le n-1$, define $\mathbf{v}_i^{(j)}$ as above. Then the span of $(m+\ell)$ vectors $\left\{\left(\mathbf{v}_i^{(j(r+1))}\right)_{i=0}^{m+\ell-1} : 0 \le j \le m+\ell-1\right\}$ is the same as the span of the $(m+\ell)$ vectors

$$\mathbf{J} \triangleq = \{(\mathbf{j}_{r+1}, \mathbf{0}, \ldots, \mathbf{0}), (\mathbf{0}, \mathbf{j}_{r+1}, \ldots, \mathbf{0}), \ldots, (\mathbf{0}, \mathbf{0}, \ldots, \mathbf{j}_{r+1})\}.$$

Proof: For $0 \le j \le m+\ell-1$ and $0 \le i \le m+\ell-1$, we observe that

$$\mathbf{v}_i^{j(r+1)} = ((\omega^i)^{j(r+1)}, (\omega^i\alpha)^{j(r+1)}, \ldots, (\omega^i\alpha^r)^{j(r+1)}) = (\beta^{ij}, \beta^{ij}, \ldots, \beta^{ij}) = \beta^{ij}\mathbf{j}_{r+1},$$

where $\beta = \omega^{r+1}$.

Therefore,

$$
\begin{pmatrix}
\mathbf{v}_0^{(0)} & \mathbf{v}_1^{(0)} & \cdots & \mathbf{v}_{m+\ell-1}^{(0)} \\
\mathbf{v}_0^{(r+1)} & \mathbf{v}_1^{(r+1)} & \cdots & \mathbf{v}_{m+\ell-1}^{(r+1)} \\
\vdots & \vdots & \ddots & \vdots \\
\mathbf{v}_0^{(m+\ell-1)(r+1)} & \mathbf{v}_1^{(m+\ell-1)(r+1)} & \cdots & \mathbf{v}_{m+\ell-1}^{(m+\ell-1)(r+1)}
\end{pmatrix}
$$

$$
= \begin{pmatrix}
\mathbf{j}_{r+1} & \mathbf{j}_{r+1} & \cdots & \mathbf{j}_{r+1} \\
\mathbf{j}_{r+1} & \beta\mathbf{j}_{r+1} & \cdots & \beta^{m+\ell-1}\mathbf{j}_{r+1} \\
\vdots & \vdots & \ddots & \vdots \\
\mathbf{j}_{r+1} & \beta^{(m-\ell-1)}\mathbf{j}_{r+1} & \cdots & \beta^{(m-\ell-1)^2}\mathbf{j}_{r+1}
\end{pmatrix}
$$

$$= \begin{pmatrix} 1 & 1 & \cdots & 1 \\ 1 & \beta & \cdots & \beta^{m+\ell-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \beta^{(m-\ell-1)} & \cdots & \beta^{(m-\ell-1)^2} \end{pmatrix} \begin{pmatrix} \mathbf{j}_{r+1} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{j}_{r+1} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{j}_{r+1} \end{pmatrix}$$

Since $\left(\beta^{ij}\right)_{0\le i,j\le m-\ell-1}$ is invertible, we have the row spaces of the two matrices to be the same, as desired. ■

Next, we demonstrate that the $\mathbf{H}$ has full rank. In other words, the rank of $\mathbf{H}$ is $m+\ell+\ell r$. Notice that the row space of $\mathbf{H}$ is given by the span of $\mathbf{J} \cup \left\{ \left(\mathbf{v}_i^{(j)}\right)_{i=0}^{m+\ell-1} : 0 \le j \le \ell r + \ell - 1, r+1 \nmid j \right\}$. From Lemma 3.4, this space is given by $m+\ell+\ell r$ vectors from $\mathbf{V}$. Since these n vectors are linearly independent, we have $\mathbf{H}$ has full rank as desired.

Therefore, the code $\mathcal{C}$ with parity-check matrix $\mathbf{H}$ has the desired dimension k. Let us partition the n coordinates into $m+\ell$ groups of size $r+1$. Then from the first $m+\ell$ rows of $\mathbf{H}$, we see any codeword in $\mathcal{C}$ has the property that the symbols in each group sum to zero. Hence, $\mathcal{C}$ has all-symbol locality r with addition based repair.

It then remains to verify the distance of $\mathcal{C}$. Let $\mathbf{H}'$ be the $(\ell + \ell r + 1) \times n$ matrix that consists of the rows $\left\{ \left(\mathbf{v}_i^{(j)}\right)_{i=0}^{m+\ell-1} : 0 \le j \le \ell r + \ell \right\}$. It follows from Lemma 3.4 that the code generated by $\mathbf{H}$ contains the code generated by $\mathbf{H}'$. So, $\mathcal{C}$ is contained in the code with parity-check matrix $\mathbf{H}'$.

On the other hand, $\mathbf{H}'$ is a generator matrix of a Reed-Solomon $[n, \ell + \ell r + 1, n - \ell - \ell r]$ code. Since the dual of an MDS code is also MDS, the code with parity-check matrix $\mathbf{H}'$ is a $[n, n-\ell-\ell r-1, \ell+\ell r+2]$ code. Therefore, the code $\mathcal{C}$ has distance $\ell + \ell r + 2 = t + 2$, as desired.

Example 3.2. Let $k=6$, $r=3$, and $n=12$. Choose $q=13$ so that $r+1|q-1$ and $n|q-1$. Then $\mathbf{H}$ is given by

$$\left(\begin{array}{cccc|cccc|cccc} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 8 & 12 & 5 & 2 & 3 & 11 & 10 & 4 & 6 & 9 & 7 \\ 1 & 12 & 1 & 12 & 4 & 9 & 4 & 9 & 3 & 10 & 3 & 10 \\ 1 & 5 & 12 & 8 & 8 & 1 & 5 & 12 & 12 & 8 & 1 & 5 \end{array}\right),$$

and the linear code whose parity-check matrix is $\mathbf{H}$ is given in Example 1.1.

Remark 3. We observe certain similarities with the class of *Tamo-Barg* codes, proposed by Tamo and Barg [8]. The Tamo-Barg codes may be viewed as a generalization of the Reed-Solomon codes. As with Reed-Solomon code, each codeword in a Tamo-Barg code is the evaluation of a low degree polynomial over a subset of points in $\mathbb{F}_q$.

Suppose[1] $r+1|n$. In general, a Tamo-Barg code requires a collection of $n/(r+1)$ disjoint groups $S_0, S_1, \ldots, S_{n/(r+1)-1}$ of evaluation points in $\mathbb{F}_q$ and a polynomial $g(x)$ of degreee $r+1$ such that g is constant on the points in the same group. Using g, Tamo and Barg then choose the space of polynomials and evaluate them onto the points in $\bigcup_{i=0}^{n/(r+1)-1} S_i$ to obtain the codewords. In some sense, Tamo and Barg constructs a *generator matrix* using g and S_s.

In Construction II, we also have $n/(r+1) = m+\ell$ disjoint groups $S_i = \{\omega^i \alpha^j : 0 \le j \le r\}$, $0 \le i \le m+\ell-1$, and we may choose $g(x) = x^{r+1}$. However, a fundamental difference in our approach is that we exploit g and S_i to construct a *parity-check matrix* for our codes. Furthermore, if we apply Tamo and Barg's construction with the parameters $n=12$, $q=13$, $k=6$, $r=3$, we do not obtain a code with addition based repair.

4. Comparison with Previous Work

In this section, we compare the family of codes constructed in Theorem 2.3 and 2.5 with certain families of optimal codes with information locality r. Here, we focus on codes defined over fields of size $\mathcal{O}(n)$, where n is the code length. As pointed out in Remark 1(ii), when the underlying field is large, the complexity of finite field arithmetic is appreciably increased.

[1] The condition of $r+1|n$ may be relaxed (see for example Constructions 5 and 6 in [8]). However, we consider the case of $r+1|n$ for ease of exposition.

TABLE II
COMPARISONS OF CODES WITH GOOD LOCALITY AND DISTANCE PROPERTIES

Codes of length n, dimension k and information locality $0 < r < k \leq n$. Assume also, $r|k$ and let $t = n - k - k/r$.

	Repair Complexity	Locality	Distance
Theorem 2.3	$\mathcal{O}(r)$ additions only for $\frac{k}{r}(r+1)$ nodes $\mathcal{O}(t)$ additions only for t nodes	$\frac{k}{r}(r+1)$ nodes with locality r t parity nodes with locality $\min\{t-1,k\}$	$t+1$
Theorem 2.5	$\mathcal{O}(r)$ additions only for all nodes	All-symbols locality r	$t+2$
Pyramid Codes	$\mathcal{O}(r)$ field operations for $\frac{k}{r}(r+1)$ nodes $\mathcal{O}(k)$ field operations for t nodes	$\frac{k}{r}(r+1)$ nodes with locality r t parity nodes with locality k	$t+2$
Tamo-Barg	$O(r^2)$ field operations for all nodes	All-symbols locality r	$t+2$

On the other hand, when the underlying field is binary, any linear code is naturally equipped with addition based repair. Binary linear codes with good locality and distance properties were first studied by Goparaju and Calderbank [15] and their work was later extended by Tamo *et al.* [16] and Zeh and Yaakobi [17]. However, optimal codes with all-symbol locality r and distance d are only known for a limited set of parameters and other exceptional cases [15], [17].

Hence, we compare our construction with the following two classes of optimal codes with information locality r: *Pyramid codes*, proposed by Huang *et al.* [18] and *Tamo-Barg codes*. Table II summarizes the differences in repair complexities, locality and distance properties, and the analysis is detailed below.

We observe that the codes from Theorem 2.3 and 2.5 are most efficient in terms of repair complexities. However, the codes from Theorem 2.3 are slightly weaker in terms of distance.

Pyramid Codes. Pyramid codes are a family of erasure codes with information locality r that attain the bound (1) and one variant is currently deployed in Windows Azure System [12]. In general, a Pyramid code is a linear $[n, k, t + 2]$ code whose k information nodes are divided into m groups $S_1, S_2, \ldots, S_m$ of size r. For each group S_i of r information nodes $x_1^{(i)}, x_2^{(i)}, \ldots, x_r^{(i)}$, we associate a parity node y whose value is given by

$$y^{(i)} = \sum_{j=1}^{r} \alpha_j^{(i)} x_j^{(i)} \text{ for some nonzero values } \alpha_1, \alpha_2, \ldots, \alpha_r.$$

Observe that to recover $x_\ell^{(i)}$, $\ell \in [r]$, we compute

$$x_\ell^{(i)} = \left(\alpha_\ell^{(i)}\right)^{-1} \left(y - \sum_{j \neq \ell} \alpha_j^{(i)} x_j^{(i)}\right).$$

In either case, to repair any of these $m(r + 1)$ nodes, we require $\mathcal{O}(r)$ finite field operations that include multiplications.

For each of the remaining t parity nodes, say z_ℓ, $\ell \in [t]$, its value is given by

$$z_\ell = \sum_{i=1}^{m} \sum_{j=1}^{r} \beta_j^{(i)} x_j^{(i)} \text{ for some } k \text{ nonzero values } \beta_j^{(i)}.$$

Hence, the repair of each of these t parity nodes require $\mathcal{O}(k)$ finite field operations that include multiplications.

Tamo-Barg Codes. Suppose $r + 1|n$. Recall that a Tamo-Barg code requires a collection of $n/(r + 1)$ disjoint groups $S_0, S_1, \ldots, S_{n/(r+1)-1}$ of evaluation points in $\mathbb{F}_q$. Then the repair function for node $\alpha \in S_s$ is given by

$$c_\alpha = \sum_{\beta \in S_s \setminus \{\alpha\}} c_\beta \prod_{\beta' \in S_s \setminus \{\alpha, \beta\}} \frac{\alpha - \beta'}{\beta - \beta'}.$$

Therefore, the repair of node α requires $\mathcal{O}(r^2)$ finite field operations that include multiplications.

5. Concluding Remarks

We introduced the notion of addition based repair for locally repairable codes in order to accelerate the repair process for single node failures. Furthermore, since the repair involves only addition operations, implementation of these codes at the hardware level is considerably easier.

The theoretical results are encouraging. We constructed two families of codes with information locality r and addition based repair, whose distances come close to the Singleton-like bound (1). In addition, the codes in Construction II bear a resemblance to the Tamo-Barg codes (see Remark 3) and it will be interesting to derive certain connections between the two code families.

Another direction of study is to allow other efficient field operations, such as the bit-wise shift operation for binary extension fields, in the repair process. As in the study of array codes [9] and regenerating codes [10], [11], the introduction of these operators may be employed to improve the encoding and decoding complexities for codes with local repair.

References

[1] P. Gopalan, C. Huang, H. Simitci, and S. Yekhanin, "On the locality of codeword symbols," *IEEE Trans. Inf. Theory*, vol. 58, no. 11, pp. 6925–6934, 2012.

[2] F. Oggier and A. Datta, "Self-repairing homomorphic codes for distributed storage systems," in *2011 Proc. IEEE Int. Conf. Computer Comm. (INFOCOM)*, 2011, pp. 1215–1223.

[3] D. S. Papailiopoulos and A. G. Dimakis, "Locally repairable codes," in *Proc. IEEE Symp. Inf. Theory (ISIT)*, 2012, pp. 2771–2775.

[4] N. Silberstein, A. S. Rawat, O. O. Koyluoglu, and S. Vishwanath, "Optimal locally repairable codes via rank-metric codes," in *Proc. IEEE Symp. Inf. Theory*, 2013, pp. 1819–1823.

[5] I. Tamo, D. S. Papailiopoulos, and A. G. Dimakis, "Optimal locally repairable codes and connections to matroid theory," in *Information Theory Proceedings (ISIT), 2013 IEEE International Symposium on*. IEEE, 2013, pp. 1814–1818.

[6] W. Song, S. H. Dau, C. Yuen, and T. J. Li, "Optimal locally repairable linear codes," *IEEE J. Sel. Areas Commun. (JSAC)*, vol. 32, no. 5, pp. 1019–1036, 2014.

[7] T. Ernvall, T. Westerback, and C. Hollanti, "Constructions of optimal and almost optimal locally repairable codes," in *4th Intl. Conf. Wireless Commun., Vehicular Technology, Inf. Theory, Aerospace Elect. Systems (VITAE)*. IEEE, 2014, pp. 1–5.

[8] I. Tamo and A. Barg, "A family of optimal locally recoverable codes," *IEEE Trans. Inf. Theory*, vol. 60, no. 8, pp. 4661–4676, Aug 2014.

[9] M. Blaum and R. M. Roth, "New array codes for multiple phased burst correction," *IEEE Trans. Inf. Theory*, vol. 39, no. 1, pp. 66–77, 1993.

[10] H. Hou, K. W. Shum, M. Chen, and H. Li, "Basic regenerating code: Binary addition and shift for exact repair," in *Proc. IEEE Symp. Inf. Theory (ISIT)*, 2013, pp. 1621–1625.

[11] K. W. Shum, H. Hou, M. Chen, H. Xu, and H. Li, "Regenerating codes over a binary cyclic code," in *Proc. IEEE Symp. Inf. Theory (ISIT)*, 2014, pp. 1046–1050.

[12] C. Huang, H. Simitci, Y. Xu, A. Ogus, B. Calder, P. Gopalan, J. Li, and S. Yekhanin, "Erasure coding in windows azure storage," in *Proc. USENIX Conf. Annual Technical Conf. (ATC)*, no. 2, 2012.

[13] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error-Correcting Codes*. Amsterdam: North-Holland, 1977.

[14] D. Hankerson, S. Vanstone, and A. Menezes, "Finite field arithmetic," *Guide to Elliptic Curve Cryptography*, pp. 25–73, 2004.

[15] S. Goparaju and R. Calderbank, "Binary cyclic codes that are locally repairable," in *Proc. IEEE Symp. Inf. Theory (ISIT)*, 2014, pp. 676–680.

[16] I. Tamo, A. Barg, S. Goparaju, and R. Calderbank, "Cyclic LRC codes and their subfield subcodes," *arXiv preprint arXiv:1502.01414*, 2015.

[17] A. Zeh and E. Yaakobi, "Optimal linear and cyclic locally repairable codes over small fields," *arXiv preprint arXiv:1502.06809*, 2015.

[18] C. Huang, M. Chen, and J. Li, "Pyramid codes: Flexible schemes to trade space for access efficiency in reliable data storage systems," *ACM Transactions on Storage (TOS)*, vol. 9, no. 1, p. 3, 2013.